Developing a System of Knowledge Bases Using Semantic Schemas in Java and CLIPS

Sorin Dincă
Faculty of Accounting and Financial Management Craiova
Spiru Haret University
sorin.dinca@gmail.com

Claudiu Ionuţ Popîrlan
University of Craiova, Department of Computer Science,
Craiova, Romania
popirlan@inf.ucv.ro

Abstract
In this paper a knowledge base system based on semantic schemas are developed. The basic idea of this paper is to develop a literary assistant to assign a particular book to a reader according to the age, education, and interests. The application does not pretend to be a real world system, but it could be a limited version of what it should be. Therefore, it could be used (in large-scale) in a library or bookstore to advise readers which book they should read. For this application implementation we use CLIPS [20] and Java [23].

Keywords: knowledge base system, semantic schemas
ACM/AMS Classification: 68U99

1. Introduction

Knowledge-Based Systems ([2], [7], [8]) are the result of a long investigation process performed by Artificial Intelligence scientists. They started to understand that the capabilities of a computer program to solve problems are not the formal expression or the inference logic schemes. Therefore, a Knowledge-Based System is a group of computer programs that tries to simulate a human expert in his field. The idea is to create the expertise of a competent professional, not to replace human thinking by computer program.

In this paper we develop an application based on knowledge bases ([1], [3], [4], [5], [6], [10], [11], [12], [18], [19]) with semantic schemas ([13], [14], [15], [16], [17]) for literary assistant to assign a particular book to a reader according to the age, education, and interests.
2. Application Knowledge Description

It describes the concepts and concept relationships involved in a domain (mechanic, medicine, etc.). The concept is the representative central entity in the domain knowledge. A concept is identified by its name, and refers to abstract entities (patient) or specific entities. It is like the Java class concept. The concepts are described by their properties or attributes, defined by their type and name. The property is the atom of domain knowledge representation. The domain knowledge is composed of three elements: Class Diagram, Expressions Relationship and Knowledge Base.

2.1 Class Diagram

We define the concepts and attributes using UML (Unified Modeling Language), as shown in Figure 1, although we could use other AI modeling languages like KIF (Knowledge Interchange Format).

![Figure 1. Application Class Diagram](image)

After drawing the diagram, we will write them in CML (Conceptual Modeling Language) [9]:

```
Concept Person;
Attributes:
Name: String;
Age: Int;
Education: String;
Interests: String;
```
2.1 *Expressions Relationship*

They represent relations in an *if . . . then* rule form: a conditional expression in the antecedent, and assignment expression in the consequent. They suppose a cause-effect association. In our example, the relations are presented in Figures 2, 3, 4 and 5.

![Figure 2. Expressions Relationship (a)](image)

![Figure 3. Expressions Relationship (b)](image)
3. Knowledge Base using Semantic Schemas

The knowledge-base gets the pairs through different generic rules previously defined. Shown below is a sample instance of the knowledge-base.

Knowledge-base assistant
Expressions
Person.age <= 14
ABSTRACT Person.stage = CHILD

Person.age > 14 AND Person.age <= 18
ABSTRACT Person.stage = TEENAGER
Person.age > 18 AND Person.age <= 35
ABSTRACT Person.stage = YOUNG
Person.age > 35
Person.interests = "Sport" OR Person.interests = "Painting" OR
Person.interests = "Music" OR Person.interests = "Cooking"
Person.education = SECONDARY AND Person.stage = YOUNG
SELECT-LEVEL reader.level = ADVANCED
Person.education = SECONDARY AND Person.stage = ADULT
SELECT-LEVEL reader.level = ADVANCED
Inference knowledge refers to the subtask sets that do not need subsequent decomposition. They are the reasoning primitives and the elemental reasoning steps for task resolution. They are described by specifying the performed function and their input and output. It's important to note that inference description does not imply how it is performed, because it depends strongly on the particular application and domain.

### 3.2 Inference Scheme

As explained before, the inference is the basic reasoning step, but it depends on the domain roles:

- **Static roles**: They are the domain elements that are used in the reasoning process but are not affected by it.
- **Dynamic roles**: They are the inference input and output that sign the domain elements that will be used during the reasoning process.

```plaintext
INFERENCE Abstract
ROLES
INPUT: Person;
OUTPUT: Stage;
STATIC: Abstraction-model;
SPECIFICATION:
"Abstracts the age of a person"
END INFERENCE Abstract;
INFERENCE Select-genre
ROLES
INPUT: Interest;
OUTPUT: Genre;
STATIC: Preferences-model;
```
SPECIFICATION:
"Selects a genre according to the user’s interests"
END INFERENCE Abstract;

INFERENCE Select-level
ROLES
INPUT: Education, Stage;
OUTPUT: Level;
STATIC: Level-model;
SPECIFICATION:
"Selects a level according to the user’s education and stage"
END INFERENCE Abstract;

INFERENCE Select-book
ROLES
INPUT: Genre, Level;
OUTPUT: Book;
STATIC: Book-selection-model;
SPECIFICATION:
"Assigns a book according to the user’s level and selected genres"
END INFERENCE Abstract;

3.3 Domain Connection

After modeling the inference knowledge, it is necessary to describe its connection with the elements of the knowledge domain. In our example, its results are shown in Figure 6.

![Figure 6. Domain Connection](image)

To implement the example, I chose to use CLIPS, an expert system tool with these important features:

*Knowledge representation:* CLIPS provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented, and procedural.
Integration/Extensibility: CLIPS can be embedded within procedural code, called as a subroutine, and integrated with languages such as Java.

4. Java Implementation and results

The simplest form to embed the CLIPS source into Java is to include the header file through the extern keyword in the Eclipse project [21]. Next, we will call InitializeEnvironment to run the CLIPS engine, and call Load to load the file containing the source.


If is typed an age and an education that do not have a logic match, the system will not return any suggestion.

![Figure 6. Java implementation and Results](image-url)
5. Conclusions

The development of Knowledge-Based Systems is not sufficiently stan-
dardized. KADS (Knowledge Acquisition Design System) is a results-oriented
methodology, an emerging European methodology to develop Knowledge-Based
Systems - funded as an ESPRIT project [22]. In this paper, according with
KADS, an application based on knowledge bases with semantic schemas for
literary assistant to assign a particular book to a reader according the age,
education, and interests was developed. The Java application using CLIPS
and interrogation results based on domain interest was presented.

References

1. Chein, M., Mugnier, M.L., *Graph-based Knowledge Representation- Computa-
tional Foundations of Conceptual Graphs*, Series: Advanced Information
3. Miranda, E., Zaffalon, M., *Coherence graphs*., Artificial Intelligence, Vol-
ume 173, Issue 1, 104–144 (2009)
Bases and Computational Properties of their Answer Functions*, Annals
of the University of Craiova, Mathematics and Computer Science Series
5. Popirlan, C.I., *A Java Implementation of Modeling Results About Strat-
tified Graphs*, 4th Romanian Conference on Artificial Intelligence and
Digital Communications Research Notes in Artificial Intelligence and
6. Popirlan, C.I., *A solution based on intelligent software agents to im-
prove the data searching in the contact centers*, In: Proceedings - 2011
6th IEEE Joint International Information Technology and Artificial In-
representation*, 11th World Multi-Conference on Systemics, Cybernet-
ics and Informatics, (WMSCI 2007), July 8-11, 2007, Orlando, Florida,
USA, Volume I, 92–96 (2007)
Functionality*, Research Notes in Artificial Intelligence and Digital Com-
9. Schreiber, G., Wielinga, B., Akkermans, H., Van de Velde, W., Anjew-
ierden, A., *CML: The commonKADS conceptual modelling language, A
Future for Knowledge Acquisition*, Lecture Notes in Computer Science,
Volume 867, 1–25 (1994)
10. Tandareanu, N., *Collaborations between distinguished representatives for
labelled stratified graphs*, Annals of the University of Craiova, Mathemat-


